**Introdución á lóxica**

**Cálculo de proposición**

* A linguaxe está composta por variables (representadas coas letras p, q, r, s…) e conectores (^, v, ¬, ->)
* Estúdase a **lóxica booleana**: as proposicións só poden ser verdadeiras ou falsas.
  + Unha variable, por si mesma, forma unha proposición.

**Conectores**

| Símbolo | Nome | Caso exc. | Condición |
| --- | --- | --- | --- |
| **¬**p | Non | V | p≡f |
| p **v** q | Ou | F | p≡f e q≡f |
| p **^** q | E | V | p≡v e q≡v |
| p **->** q | Implica | F | p≡v e q≡f |

**Operacións lóxicas**

* Pódense resolver mediante táboas de verdade ou analizando o caso excepcional de cada conector.
* **Tautoloxías:** Proposicións que sempre son certas. Represéntanse con T.
  + Exemplos: (p v ¬p), (p -> p), [p -> (q->p)]
* **Contradicións:** Proposicións que sempre son falsas. Represéntanse con ⊥
  + Exemplos: (p ^ ¬p), ¬(p ->p)
* **Continxentes:** Proposicións poden ser falsas ou verdadeiras.

**Equivalencias** (dende o punto de vista semántico)

* Proposicións que teñen os mesmos valores de verdade, pero que son distintas. Represéntase con p **≡** q.

**Consecuencias semánticas**

* p⊨q (q é **consecuencia semántica** de p) se cando p é verdade, q tamén.
  + Semánticamente idéntico a ->.
* As proposicións (T⊨p) , (⊥⊨p) son tautoloxías.
* A proposición (⊨P) é semánticamente idéntica a (P)
* Dúas proposicións son equivalentes se son consecuencia semántica unha da outra
* A proposición (p ⊨ q -> r) é semánticamente idéntica a (p^q ⊨ r) ou (p,q ⊨ r)

**Árbores semánticas**

* Esquema no que a proposiciones ‘e’ son parte da mesma rama e as proposicións ‘ou’ son bifurcacións. Cúmplense se toda unha rama é certa.
* Xeralmente, compróbase se o contrario da proposición é unha contradición.
* Exemplo:

¬a, avb ⊨ b ?

¬a ^ avb ⊨ b

⊨ [¬a ^ avb] -> b

¬ [¬a ^ avb] -> b

|

¬b

|

¬a

avb

/ \

a b

x x

* Entón, é unha contradición, polo que a orixinal proposición é unha tautoloxía.

**Cálculo de predicados**

**Elementos da linguaxe**

* Formada por **variables**(x,y,z), **funcións**(f,g,h) e **predicados**(P,Q,R).
* Ademais, conectores (v,^,¬, ->) e cuantificadores (∀, ∃)
  + Exemplo: ‘Todo número x é menor ca o número seguinte de x.'
    - ‘Todo número x’ (∀x) é un cuantificador e unha variable.
    - ‘Menor que’ é un predicado binario.
    - ‘O número seguinte de x’ é unha función que toma x como variable.
* Os predicados e funcións están asociados coa **aridade:** o número de termos que requiren. a≥0.
  + As funcións 0-arias son constantes
* Unha función co número adecuado de termos forma un novo termo (por exemplo s(x), o seguinte de x, é un termo)
* Se existen fórmulas α e ß, α^ß, αvß, ¬α e α->ß son tamén fórmulas
* Se existe unha fórmula α e unha variable x, as expresións **∀xα** e **∃xα** son tamén fórmulas.

**Subespacios de unviersos**

* Dada unha interpretación fixa das fórmulas, podemos determinar o subespazo dun universo que contén os elementos que cumplen esa fórmula.
* O universo analizado serán os números naturais U(0,1,2,3…)
* **Exemplo:** <2 é un subespacio do U2, que inclúe pares de números tales que x<y (0,1) (4,7)..
  + Entón, <(x,y) é certo se (x,y) pertenecen a <2.
  + <(x,sx) non é certa nin falsa porque as variables x non foron interpretadas
  + U⊧∀x<(x,sx) é o mesmo que U⊧<(x,sx)[toda interpretación de x], e é certa
  + Exemplo: U⊧∀x<(0,x) é falso, existe un contraexemplo en x=0

**Exemplos de fórmulas** nun determinado universo

* Funcions: C L O P (cada unha representa unha provinza de orixe), =(son iguais), A(son amigos)
* ‘Todos os que sean de Pontevedra teñen un amigo que non é de Ourense’: U⊧ ∀x[(P(x) -> ∃ y (A(x,y) ^ ¬O(y)]
* ‘Todos teñen un amigo’: U⊧ ∀x∃y A(x,y)
  + Isto non significa que x sexa distinto de y (pode ser amigo de un mesmo)
* ‘Todos teñen **só** un amigo’: U⊧ {∀x∃y A(x,y)} ^ {∀Z[A(x,z) -> z=y]}
  + Para todo x existe un y tal que son amigos. Ademáis, todo Z que sexa amigo de x é igual a y.

**Veracidade de fórmulas** en calqueira universo[[1]](#footnote-0)

* U⊧∀x∃α(y) ≢ ∃y∀α(x) (se para todo x existe algún y, non significa que exista algún y para o que existan todo x)
* U⊧∃y∀α(x) ≡ ∀x∃α(y) (se algún y se cumple todo x, significa que para todo x se cumple algún y)
* U⊧∀x∃y <(y,x) ‘todo número ten un número menor ca el’ só é certo nun universo que inclúa números negativos.
* U⊧∀x∃y [<(y,x) v =(x,y)] ‘todo número ten un número menor ou igual’ é certo en calqueira universo non nulo.
* U⊧∃y∀x [<(y,x) v =(x,y)] ‘existe algún número menor ou igual a todos os números’ é certo para cualquiera universo non nulo.
* U⊧{∀x (a -> b)} -> {(∀xa -> ∀xb)}
* Para demostrar que é certa, comprobamos se é posible que sea falso,
  + U⊧ ¬(∀xa -> ∀xb)
  + U⊧ ∀xa ^ ¬∀xb
  + U⊧ ∀xa ^ ∃x¬xb
  + Esto contradise con ∀x (a -> b). Entón, o enunciado 1 é certo en todo universo.
* U⊧{(∀xa -> ∀xb)} -> {∀x (a -> b)}
  + Exemplo: dúas funcións I(x) impar e P(x) primo. U = 5,2,9
    - U⊧(∀xI(x) -> ∀xP(x)) é certa, debido a que ∀xI(x) é falsa (non todos os números son impares)
    - U⊧∀x (I(x) -> P(X)) é falsa. Existe un número impar non primo.
    - Entón, o enunciado 2 é falso. Pode ser certa a primeira proposta e falsa a segunda.
* U⊧{(∀xa v ∀xb)} -> {∀x (a v b)}
  + É certo. Se todos teñen a ou todos teñen b, todos teñen unha das dúas.
* U⊧ {∀x (a v b)} -> {(∀xa v ∀xb)}
  + Exemplo: Dúas funcións P(x) e I(x) que representan par e impar:
    - No universo dos Z, todos os números son pares ou impares. Con todo, nin todos son pares nin todos impares.
    - O enunciado 4 é falso. Pode ser certa a primeira proposta e falsa a segunda.

**Teoría/linguaxe de conxuntos**

**Conceptos básicos**

* ∈ é un predicado binario. x∈X significa ‘x **pertenece** a X’
* ⊆ é outro predicado binario. X⊆Y significa ‘X está contido en Y’, ou X é un **subconxunto** de Y, é dicir, todos os elementos de X pertencen tamén a Y.
  + X⊆Y ^ Y⊆X ⇔ X=Y.
  + X⊂Y emprégase para denotar que X está contido pero **é distinto** de Y.
* X U Y é o conxunto formado polos elementos (t⋲x v t⋲y) (**unión**)
  + A ∩ (BUC) = (A U B) ∩ (A U C)
* X ∩ Y é o conxunto formado polos elementos (t⋲x ^ t⋲y). (**intersección**)
  + A U (B∩C) = (A ∩ B) U (A ∩ C)
* Ø é o **conxunto baleiro**. Está contido en todos os conxuntos (Ø⊆A para todo A)
  + **Demostración**: ∀x(x€Ø -> x€A). Esto é certo porque x€Ø é falso.
* A\B se x€A\B || x€A^¬x€B (A - B, conxunto **complementario** de B en A)
* P(A) “**partes** de A”: {X | X⊆A}
  + Sendo A={1,2}, P(A) = {ф, {1}, {2}, {1,2}}
* Nos conxuntos, o orde dos elementos e as veces que aparezan repetidos son **irrelevantes**.

**Notación de comprensión**

* A = {x1, x20 ,..., xk} ‘extensión’
* A = {x | ∝(x)} ‘comprensión’ (x **tal que** se cumple ∝(x))
* AxB = {(x,y) | x€A ^ y€B} (**producto cartesiano**)
  + Defínese como o conxunto de pares de elementos, tales que o primeiro pertence a A e o segundo a B.

**Exemplos de demostración en Conxuntos**

* A\(B^C) = (A\B)v(A\C)
  + A\(B^C) => x€A e non X€(B^C).
  + (A\B)v(A\C) => x€A ^¬x€B v x€A^¬x€c
  + x€A está en todos os membros e considérase comprobado.
  + ¬X€(B^C) => ¬x€B v¬x€C. É a mesma condición que a de (A\B)v(A\C).
* P(A) U P(B) ⊆ P(AUB)
  + x€P(A)UP(B) || X€P(A) v X€P(B)
  + X€P(A) v X€P(B) || X⊆A v X⊆B
  + X⊆A || X⊆AUB, X⊆B || X⊆AUB
  + Entón, en calqueira suposto, X€P(AUB), polo que se cumple a proposta.
* P(AUB) ⊆ P(A) U P(B)
  + Sendo A = {1} e B = {2}
  + P(AUB) = {O, {1}, {2}, {1,2}}
  + P(A) U P(B) = {O, {1}, {2}}
  + Logo, a proposta é **falsa.**
* P(A) ^ P(B) = P(A^B)
  + Todos os elementos que son partes tanto de A como de B serán partes de ambos.
  + Todos os elementos que son partes da intersección de A e B pertencen tanto a A como a B.

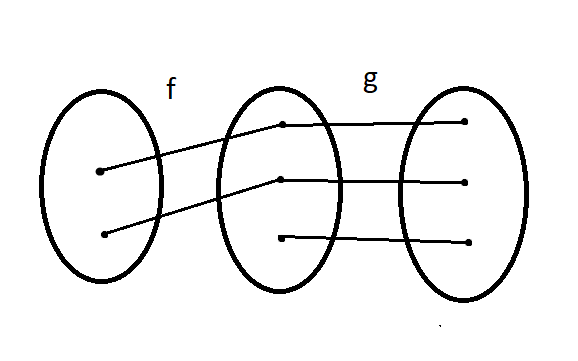
**Aplicación**

* f: A—>B ‘f é **aplicación** de A a B’. f convierte elementos de A en elementos de B.
* x€A → f(x)€B
* **Unicidade**: ∀x ∀y1 ∀y2 ( f(x)=y1 ^ f(x)=y2, y1 = y2) (todo elemento ten unha sóa imaxe)
* **Universalidade**: ∀x (x€A → Ǝy y€B, y=f(x)) (todos os elementos teñen imaxe)
* Ejemplo: f: Z —> Z, f(x) = x2
* Ejemplo: f: Z —> Z, f(x) = √x. Non é universal, non todo enteiro ten raíz enteira. Logo, non é unha aplicación. Tampouco ten unicidade.
* Pode existir unha aplicación Ø→X, pois non habería elementos desasignados.
  + Unha aplicación Ø→Ø é bixectiva.

**Tipos de aplicacións**

* f: A—> B **‘Inxectiva’** cando ∀x1∀x2 ( f(x1) = f(x2) → x1 = x2)
  + Significa que cando dous números teñen a mesma imaxe, teñen que ser idénticos. É dicir, cada imaxe é única.
* f A—> B **‘Sobrexectiva’** cando ∀y (y€B → Ǝx(x€A ^ f(x) = y))
  + Significa que todos os elementos de B son imaxe dalgún de A.
  + As funcións que son inxectivas e sobrexectivas son **bixectivas** de dúas funcións.

**Composición**

* Se f:A—>B e g: B—>C, **(gºf):A—>C** é a **composición** de f e g..
  + (gºf)(A) = g(f(A))
* Se f e g son inxectivas, (gºf) tamén é inxectiva.
  + Ǝa1 Ǝa2, (gºf)(a1) = (gºf)(a2) ^ a1 =/= a2 (comp. non inxectiva)
  + g(f(a1)) = g(f(a2) -> f(a1) = f(a2) (g inxectiva)
  + f(a1) = f(a2) → a1 = a2 (f inxectiva).
    - Isto contradise coa primeira proposición. Logo, é imposible que se f e g son inxectivas, non o sea a composición.
* Se f e g son sobreyectivas, (gºf) tamén é sobreyectiva.
* Se (gºf) é inxectiva, **f** é inxectiva. (g non ten por que selo)
  + f(a1) f(a2) —> a1=a2 (hipótese, asúmese certa para o próximo paso)
  + g(f(a1)) = g(f(a2)) → (gºf)(a1) = (gºf)(a2)
  + (g non inxectiva, f e fºg si)
* Se (gºf) é sobreyectiva, entón **g** é sobreyectiva. (f non ten por que selo).
  + (f non sobreyectiva, g e fºg si)

**Contar**

* **Contar** un conxunto A é establecer unha aplicación bixectiva entre A e as partes do conxunto dos números naturais { {}, {1}, {1,2}, {1,2,3} } (todos os numeros naturais)
* O 0 correspóndese con {}.
* Exemplo: A={a1, a2, … an}, {1,2,3…n}
  + f: a1→1, a2 →2, …, an→n. Logo, o conxunto mide ‘n’.
* Baixo esta definición, é posible contar o conxunto de numeros naturais, pois poderíase establecer unha aplicación bixectiva entre ese conxunto e si mesmo.
  + Non se poden contar os numeros reais.
* Pódese establecer unha aplicación inxectiva entre o conxunto de números naturais {0,1,2,3…} e o conxunto de pares naturais {0,2,4,6…} tal que f(n)=2n. Entón, ambos conxuntos **miden o mesmo**.
* Tamén se pode establecer unha ap.inx. entre o conxunto de números enteiros {...-3,-2,-1,0,1,2,3…} e os naturais {0,1,2,3…} Entón, os números enteiros son contables e o conxunto **mide o mesmo** ca o dos naturais.
  + Neste caso, a función que os une é f(n) =
    - n/2 se n par
    - -(n+/1)/2 se n impar
* Un conxunto A é **infinito** se existe B⊂A e f: B—>A bixectiva. É dicir, se se pode poñer en bixección cunha parte del.

**Conxunto de aplicacións**

* Dados dous conxuntos A,B, **BA = { f: A—>B}**. (conxuntos de todas as aplicacions de A en B)
  + **|BA| = |B||A|**
    - [[2]](#footnote-1)Se A={a1, …, a**n**}, B = {0,1}, BA =2n

**Principio de demostración por indución matemática**

* Método para comprobar ∀n P(n), sendo P un predicado.
  + Para isto, compróbase primeiro que P(0). (paso base)
  + Logo, compróbase que se k cumple a propiedade [hipótese de indución], o seguinte de k tamén (p(k) → p(k+1) ) (paso indutivo)
* Isto demostra que se cumple para todo n.

**Exemplos de demostración por indución matemática**

* **Exemplo:** Sumar os números 2+4+6+ … 2\*n = Sn
  + S0 = 0
  + S1 = 2 (=1x2)
  + S2 = 6 (=2x3)
  + Logo, Sn = n(n+1). Comprobamos esta hipótese mediante indución.
  + Paso base: S0 = 0x1. Certo
  + Hipótese de indución: Sk = k\*(k+1)
  + Paso indutivo: Sk+1 = (k+1)\*(k+1 +1)?
    - Sk+1 = Sk + 2(k+1) = k\*(k+1) + 2k+2
    - (k+1)(k+2) = k(k+1) + 2k + 2
    - k2 +3k +2 = k2 +3k +2.
* **Exemplo 2:** Conxeturar unha fórmula para a suma das sucesivas potencias de 2 e demostrala por indución matemática.
  + Pn = 2^0 + 2^1 + … + 2^n
    - P0 = 1
    - P1 = 1
    - P2 = 7
    - P3 = 15
    - Pn = 2^(n+1) -1
  + Paso base: P0 = 2^1 -1 = 1
  + Hipótese de indución: Pk = 2^(k+1) -1
  + Paso indutivo: Pk+1 = 2^(k+2)-1?
    - Pk+1 = Pk + 2^(k+1)Pk+1 = 2^(k+1) - 1 + 2^(k+1) = 2\*2^(k+1) -1 = 2^(k+2)-1
* **Exemplo 3:** Comparación de n^2 e 2^n.

| n | n^2 | 2^n |
| --- | --- | --- |
| 3 | 9 | 8 |
| 4 | 16 | 16 |
| 5 | 25 | 32 |

* + Conxetura: Vn(n>=4 → 2^n >= n^2)Entón, paso base será n=4. 2^4>=4^2.
    - * Paso inductivo: 2^k >=k^2 → 2^(k+1) >= (k+1)^2 ?
      * 2^(k+1) = 2 \* 2^k >= 2 \* k^2
      * 2\*k^2 >= (k+1)^2 para k>= 4?
        + (k+1)^2 = k^2 + 2k +1.
  + 2\*k^2 >= k^2 2k+1 => k^2 >= 2k+1 => k(k-2)>=4 (certo para k>=4).
* **Exemplo 4:** Tendo un tablero de 2^n \* 2^n dimensións, ao que lle falta 1 casilla. Pódese encher con pezas de 2x1 casillas?
  + Paso base: n=1, taboleiro 2x2 sen unha casilla, si que colle.
  + Paso de indución: 2^k x 2^k -1 → 2^(k+1) x 2^(k+1) -1
    - O taboleiro 2^(k+1) pódese dividir en 4 cadrantes, cada un medindo 2^k \* 2^k. Un deles terá o oco.
    - Colocamos a peza ocupando unha esquina de cada un dos cadrantes, excepto o que ten o oco.
    - Enton, quedamos con 4 cadrantes aos que lles falta un oco, sendo o mesmo que o exemplo 2^k x 2^k pero repetido 4 veces

2k .

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

.Entón, o feito de que podamos resolver un dos cadrados de 2^k implica sempre ca o cadrado de 2^(k+1) é tamén resolvible. Esto demostra o paso indutivo.

**Exemplo 5:** Demostrar que se |A| = n, |P(A)| = 2^n.

* + Paso base: n=0. |A|= 0, A = ∅, P(A) = 1
  + Hipótese de indución:[|A| = k → |P(A)| = 2^k] → [|B| = k+1 → |P(B)| = 2^(k+1)]
  + |B| = k+1 → |P(B)| = 2^(k+1) ?
  + |B| = k+1 >0
    - B = {n, a1, …, ak}
    - X€P(B) **| {**XCB | n!€X} | = 2^k. X son os subconxuntos de B que non conteñen a n. |X| = 2^k
    - X C= {a1, … ak}.
    - {y C= B | n € Y} Y son os subconxuntos de B que poden conter a N.
      * Y = {Y\{n}} U {x}. Son os elementos que non conteñen a n unidos con n. Entón, ten o mesmo número de elementos que X. |{y C= B | n € Y}| = 2^k
      * Y = {n}UZ. Z C{a1, …, ak)
  + |B| = Y + X = 2^k + 2^k = 2^(k+1)

1. a e b denotan funcións. ‘xa’ lease a(x), ∀x (a -> b) denota ∀x (a(x) -> b(x)), é dicir, todo x que cumpla a(x) cumple tamén b(x). [↑](#footnote-ref-0)
2. Cadeas binarias de n bits. [↑](#footnote-ref-1)